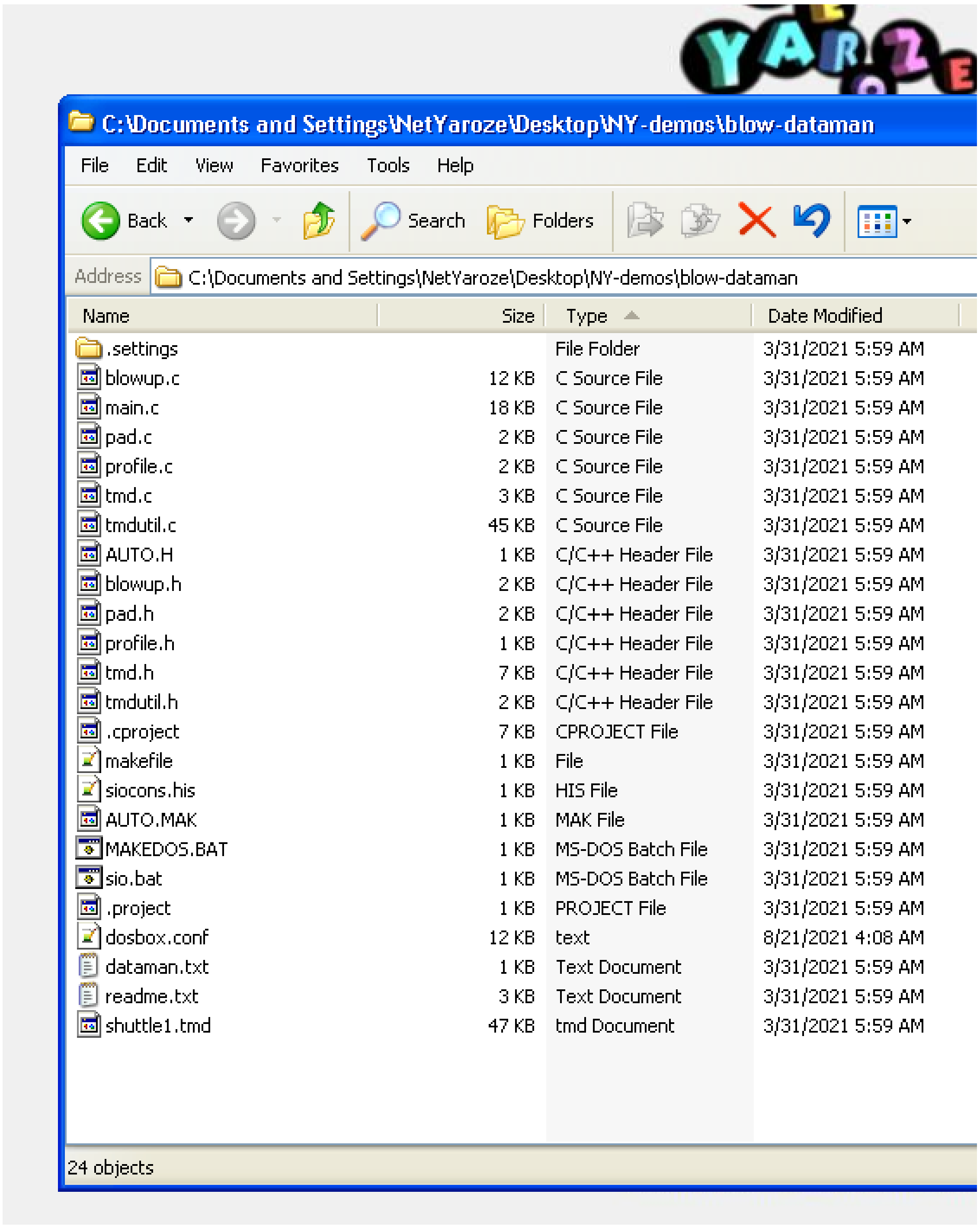
CodeWorrior Tips

Table of Contents

Import Net Yaroze GNU GCC project into CodeWarrior

Example GNU NY project:

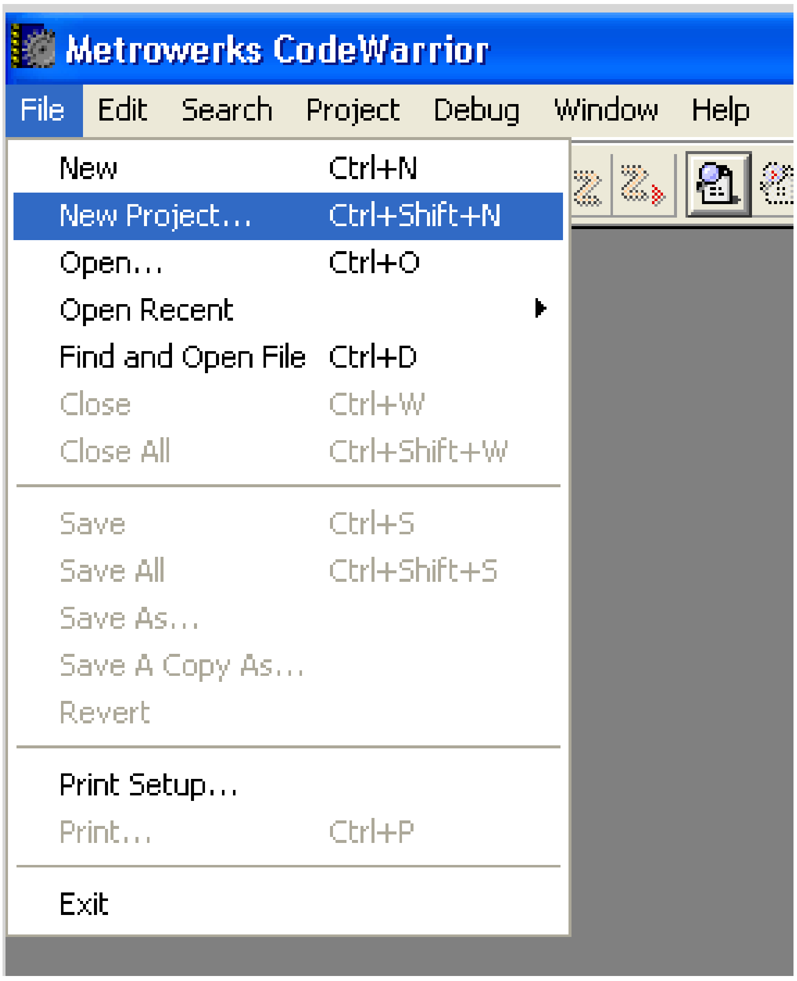

YAROZE
C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman
File Edit View Favorites Tools Help
Back
Search
Folders
Address

C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman

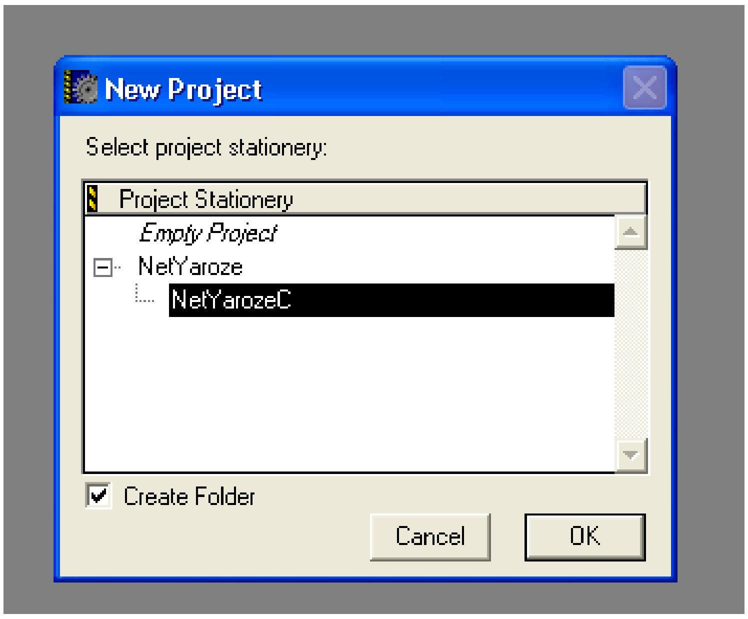
Name	Size	Type	Date Modified
.settings		File Folder	3/31/2021 5:59 AM
blowup.c	12 KB	C Source File	3/31/2021 5:59 AM
main.c	18 KB	C Source File	3/31/2021 5:59 AM
pad.c	2 KB	C Source File	3/31/2021 5:59 AM
profile.c	2 KB	C Source File	3/31/2021 5:59 AM
tmd.c	3 KB	C Source File	3/31/2021 5:59 AM
tmdutil.c	45 KB	C Source File	3/31/2021 5:59 AM
AUTO.H	1 KB	C/C++ Header File	3/31/2021 5:59 AM
blowup.h	2 KB	C/C++ Header File	3/31/2021 5:59 AM
pad.h	2 KB	C/C++ Header File	3/31/2021 5:59 AM
profile.h	1 KB	C/C++ Header File	3/31/2021 5:59 AM
tmd.h	7 KB	C/C++ Header File	3/31/2021 5:59 AM
tmdutil.h	2 KB	C/C++ Header File	3/31/2021 5:59 AM
.cproject	7 KB	CPROJECT File	3/31/2021 5:59 AM
makefile	1 KB	File	3/31/2021 5:59 AM
siocons.his	1 KB	HIS File	3/31/2021 5:59 AM
AUTO.MAK	1 KB	MAK File	3/31/2021 5:59 AM
MAKEDOS.BAT	1 KB	MS-DOS Batch File	3/31/2021 5:59 AM
sio.bat	1 KB	MS-DOS Batch File	3/31/2021 5:59 AM
.project	1 KB	PROJECT File	3/31/2021 5:59 AM
dosbox.conf	12 KB	text	8/21/2021 4:08 AM
dataman.txt	1 KB	Text Document	3/31/2021 5:59 AM
readme.txt	3 KB	Text Document	3/31/2021 5:59 AM
shuttle1.tmd	47 KB	tmd Document	3/31/2021 5:59 AM

24 objects

Start CodeWarrior's IDE and select New Project...

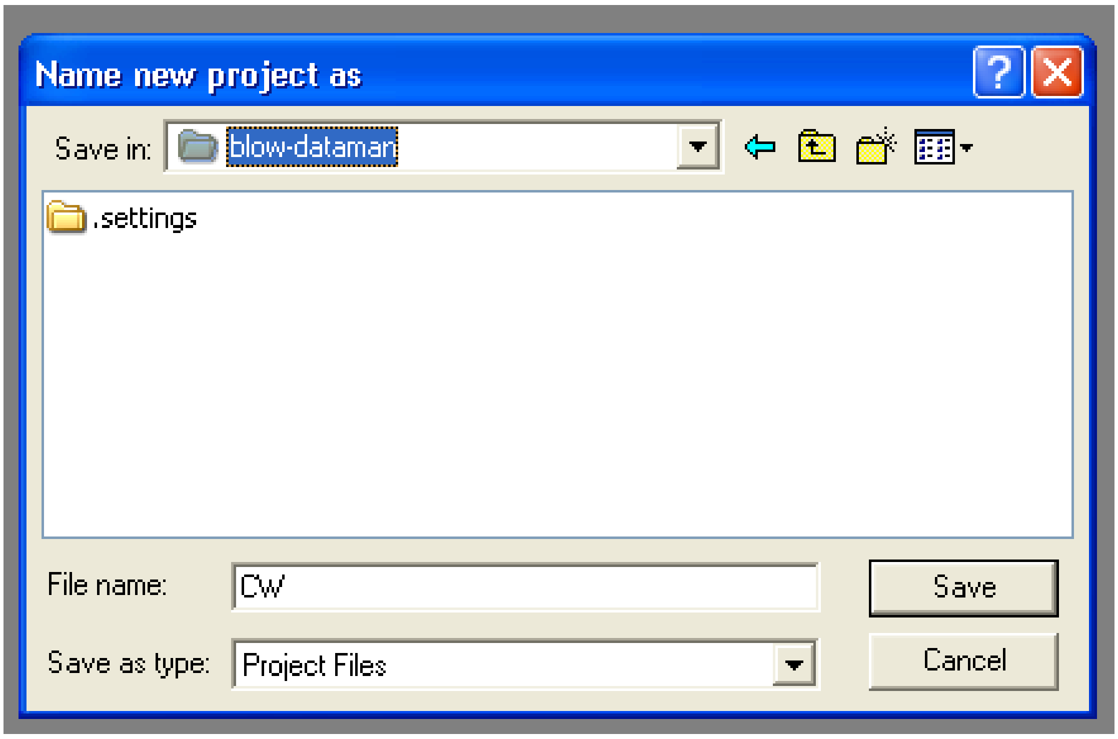


Select NetYarozeC, I like keeping CW files separate to the GNU code base, so I tick 'Create Folder', if you don't, don't tick it. Click OK to continue.



Go to the folder of the Yaroze GNU GCC project, the file name is the folder (if you ticked create fold) which CW will create it’s files in, CW is simple and clear, and this “CW” folder is used here on in.

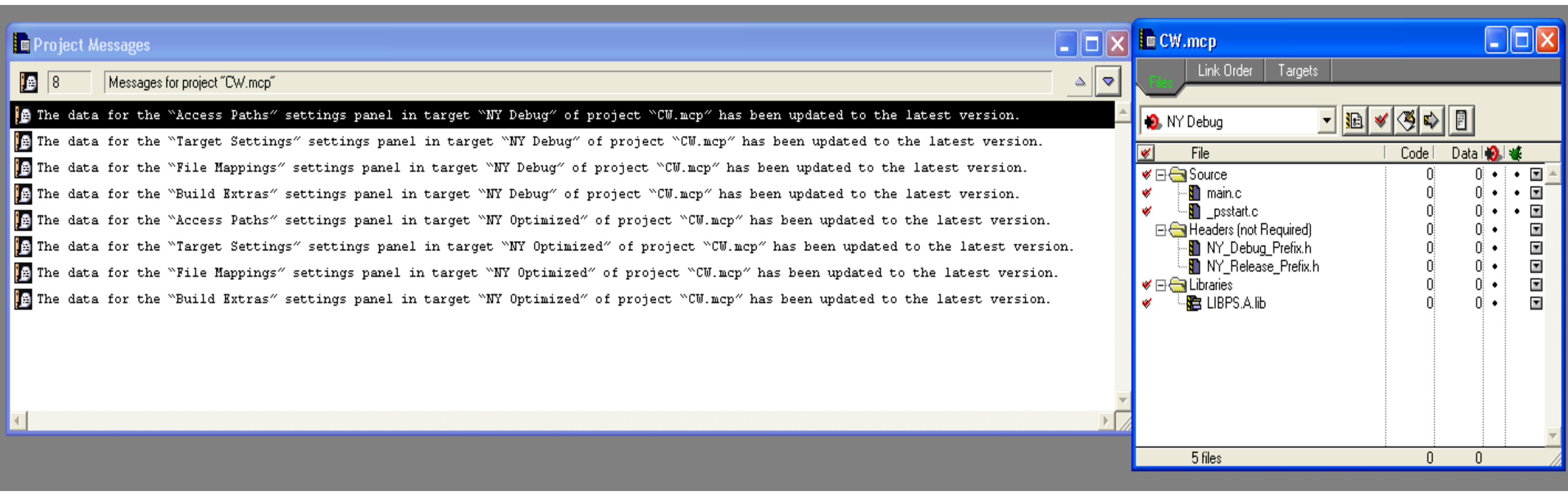
If you have multiple projects, call it something meaningful, but DON’T USE SPACES ETC!



File name: CW

Save as type: Project Files

You’ll get the following windows: left messages of the files and settings automatically applied, the right screen is the project tree with some files automatically created.

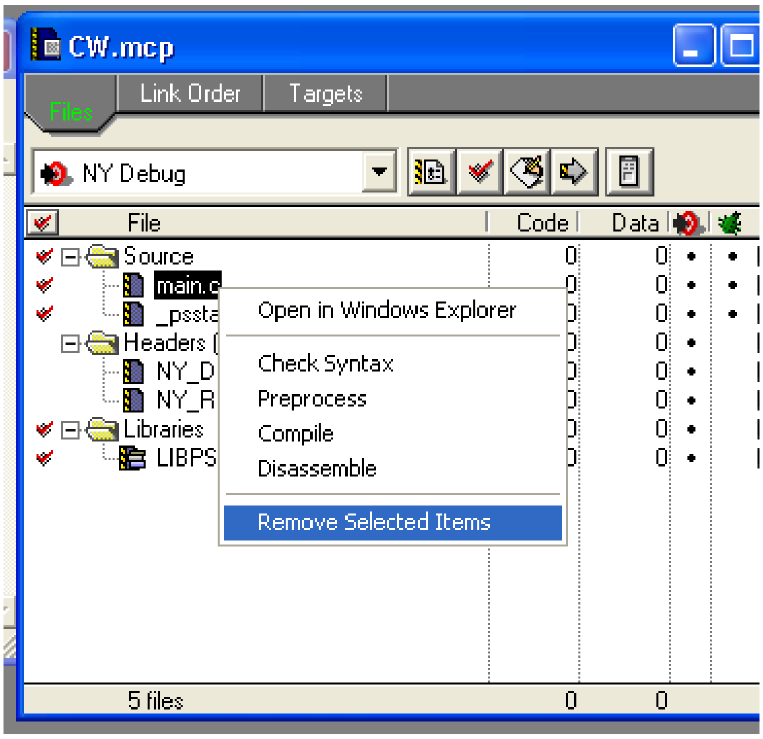


```
The data for the "Access Paths" settings panel in target "NY Debug" of project "CW.mcp" has been updated to the latest version.
The data for the "Target Settings" settings panel in target "NY Debug" of project "CW.mcp" has been updated to the latest version.
The data for the "File Mappings" settings panel in target "NY Debug" of project "CW.mcp" has been updated to the latest version.
The data for the "Build Extras" settings panel in target "NY Debug" of project "CW.mcp" has been updated to the latest version.
The data for the "Access Paths" settings panel in target "NY Optimized" of project "CW.mcp" has been updated to the latest version.
The data for the "Target Settings" settings panel in target "NY Optimized" of project "CW.mcp" has been updated to the latest version.
The data for the "File Mappings" settings panel in target "NY Optimized" of project "CW.mcp" has been updated to the latest version.
The data for the "Build Extras" settings panel in target "NY Optimized" of project "CW.mcp" has been updated to the latest version.
```

File	Code	Data
Source	0	0
main.c	0	0
_psstart.c	0	0
Headers (not Required)	0	0
NY_Debug_Prefix.h	0	0
NY_Release_Prefix.h	0	0
Libraries	0	0
LIBPS.A.lib	0	0
5 files	0	0

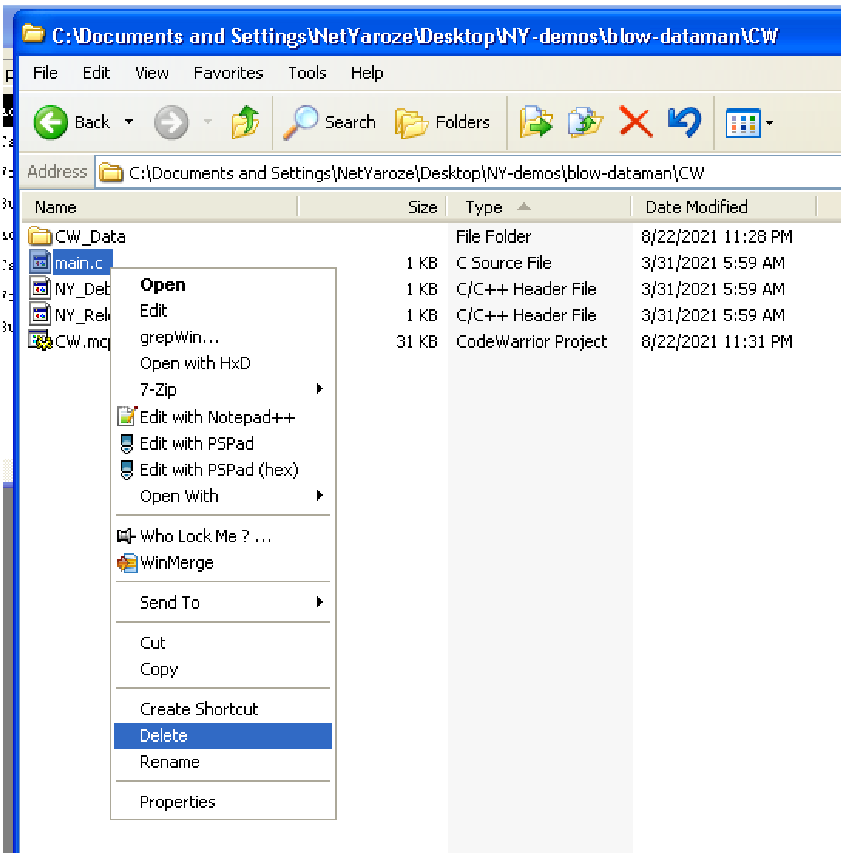
In the folder ‘Headers (not Required)’ it automatically creates 2 header files for each target with debug defines… but these are “not required” to be used.

CW will create a default **main.c** in the CW Source folder, which isn’t needed and has to be remove from the tree.



File	Code	Data
Source	0	0
main.c	0	0
_pssta	0	0
Headers (	0	0
NY_D	0	0
NY_R	0	0
Libraries	0	0
LIBPS	0	0
5 files	0	0

Now go back to the file explorer and enter the new CW created folder you should see the CW.mcp file (Metrowork Codewarrior Project) and and other files…. delete **main.c** from there also.

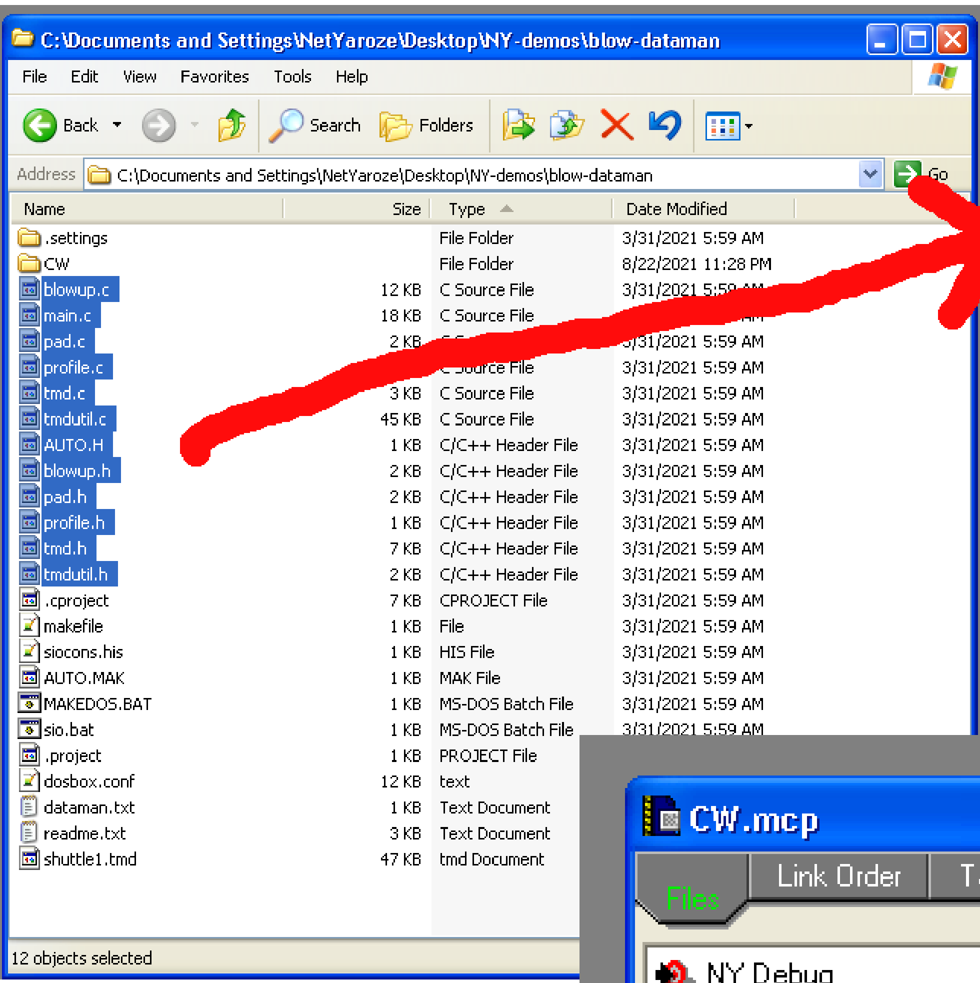


Address C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman\CW

Name	Size	Type	Date Modified
CW_Data		File Folder	8/22/2021 11:28 PM
main.c	1 KB	C Source File	3/31/2021 5:59 AM
NY_Det	1 KB	C/C++ Header File	3/31/2021 5:59 AM
NY_Rel	1 KB	C/C++ Header File	3/31/2021 5:59 AM
CW.mc	31 KB	CodeWarrior Project	8/22/2021 11:31 PM

Now go back to the NY GNU GCC project folder to drag and drop source files.

Select all source files (C/CPP and header files only) and drag and drop them to the Source folder in the CW Tree window.

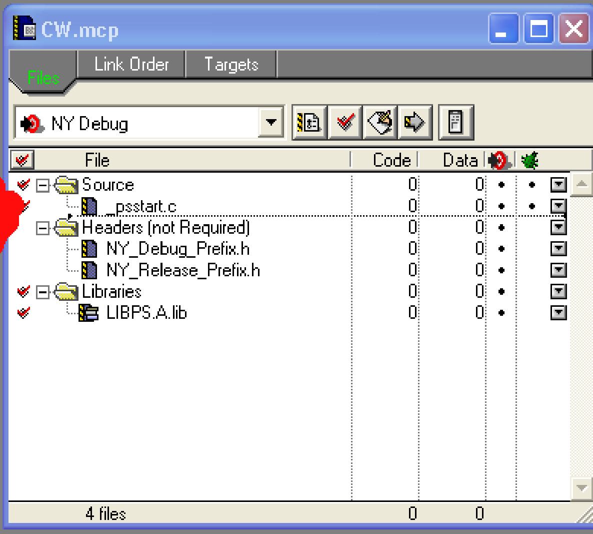


C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman

Address: C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman

Name	Size	Type	Date Modified
.settings		File Folder	3/31/2021 5:59 AM
CW		File Folder	8/22/2021 11:28 PM
blowup.c	12 KB	C Source File	3/31/2021 5:59 AM
main.c	18 KB	C Source File	[illegible]
pad.c	2 KB	[illegible]	3/31/2021 5:59 AM
profile.c	[illegible]	C Source File	3/31/2021 5:59 AM
tmd.c	3 KB	C Source File	3/31/2021 5:59 AM
tmdutil.c	45 KB	C Source File	3/31/2021 5:59 AM
AUTO.H	1 KB	C/C++ Header File	3/31/2021 5:59 AM
blowup.h	2 KB	C/C++ Header File	3/31/2021 5:59 AM
pad.h	2 KB	C/C++ Header File	3/31/2021 5:59 AM
profile.h	1 KB	C/C++ Header File	3/31/2021 5:59 AM
tmd.h	7 KB	C/C++ Header File	3/31/2021 5:59 AM
tmdutil.h	2 KB	C/C++ Header File	3/31/2021 5:59 AM
.cproject	7 KB	CPROJECT File	3/31/2021 5:59 AM
makefile	1 KB	File	3/31/2021 5:59 AM
siocons.his	1 KB	HIS File	3/31/2021 5:59 AM
AUTO.MAK	1 KB	MAK File	3/31/2021 5:59 AM
MAKEDOS.BAT	1 KB	MS-DOS Batch File	3/31/2021 5:59 AM
sio.bat	1 KB	MS-DOS Batch File	3/31/2021 5:59 AM
.project	1 KB	PROJECT File	
dosbox.conf	12 KB	text	
dataman.txt	1 KB	Text Document	
readme.txt	3 KB	Text Document	
shuttle1.tmd	47 KB	tmd Document	

12 objects selected



CW.mcp

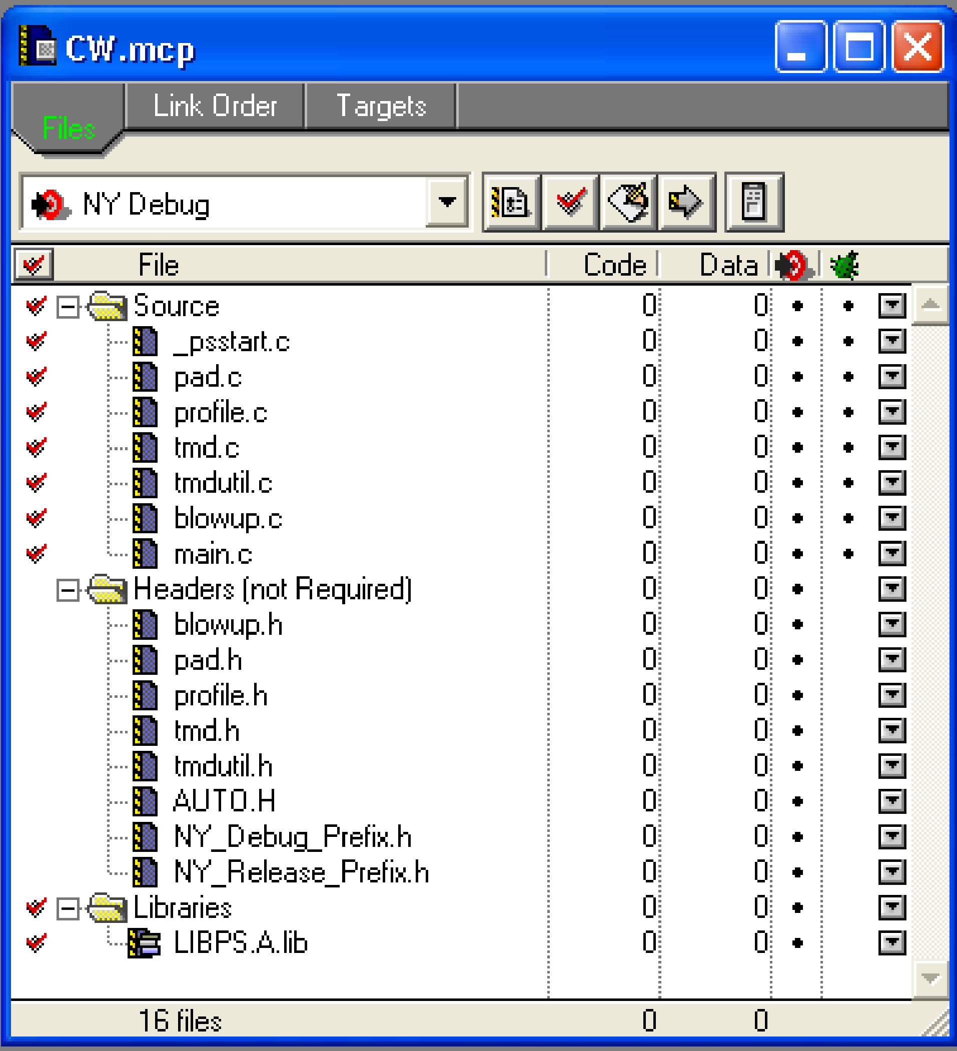
Files | Link Order | Targets

NY Debug

File	Code	Data
Source	0	0
_psstart.c	0	0
Headers (not Required)	0	0
NY_Debug_Prefix.h	0	0
NY_Release_Prefix.h	0	0
Libraries	0	0
LIBPS.A.lib	0	0
4 files	0	0

You can separate source and header files by dropping/moving C/CPP files in the source and .h files in the Headers folder.

You can change the "Headers (not Required)" by double clicking it.



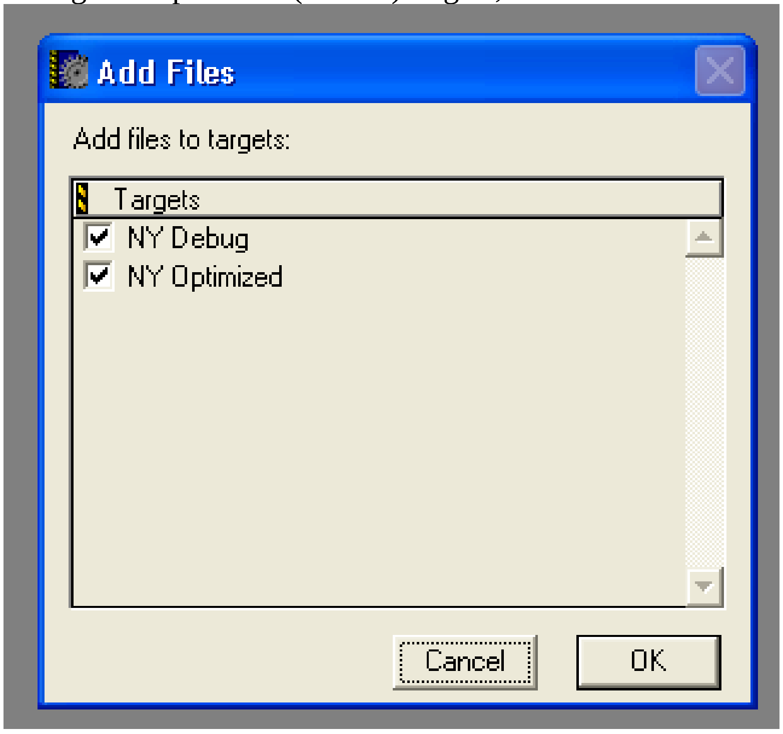
CW.mcp

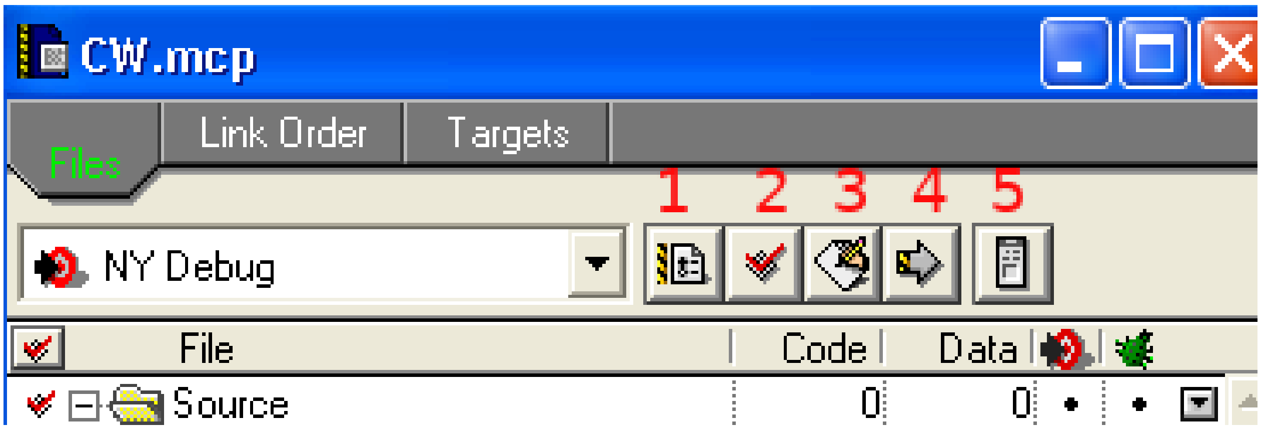
Files | Link Order | Targets

NY Debug

File	Code	Data
Source	0	0
_psstart.c	0	0
pad.c	0	0
profile.c	0	0
tmd.c	0	0
tmdutil.c	0	0
blowup.c	0	0
main.c	0	0
Headers (not Required)	0	0
blowup.h	0	0
pad.h	0	0
profile.h	0	0
tmd.h	0	0
tmdutil.h	0	0
AUTO.H	0	0
NY_Debug_Prefix.h	0	0
NY_Release_Prefix.h	0	0
Libraries	0	0
LIBPS.A.lib	0	0
16 files	0	0

When you add files, you will be asked what targets are the files for, typically you want all the source file for both Debug and Optimized (release) targets, click ok.





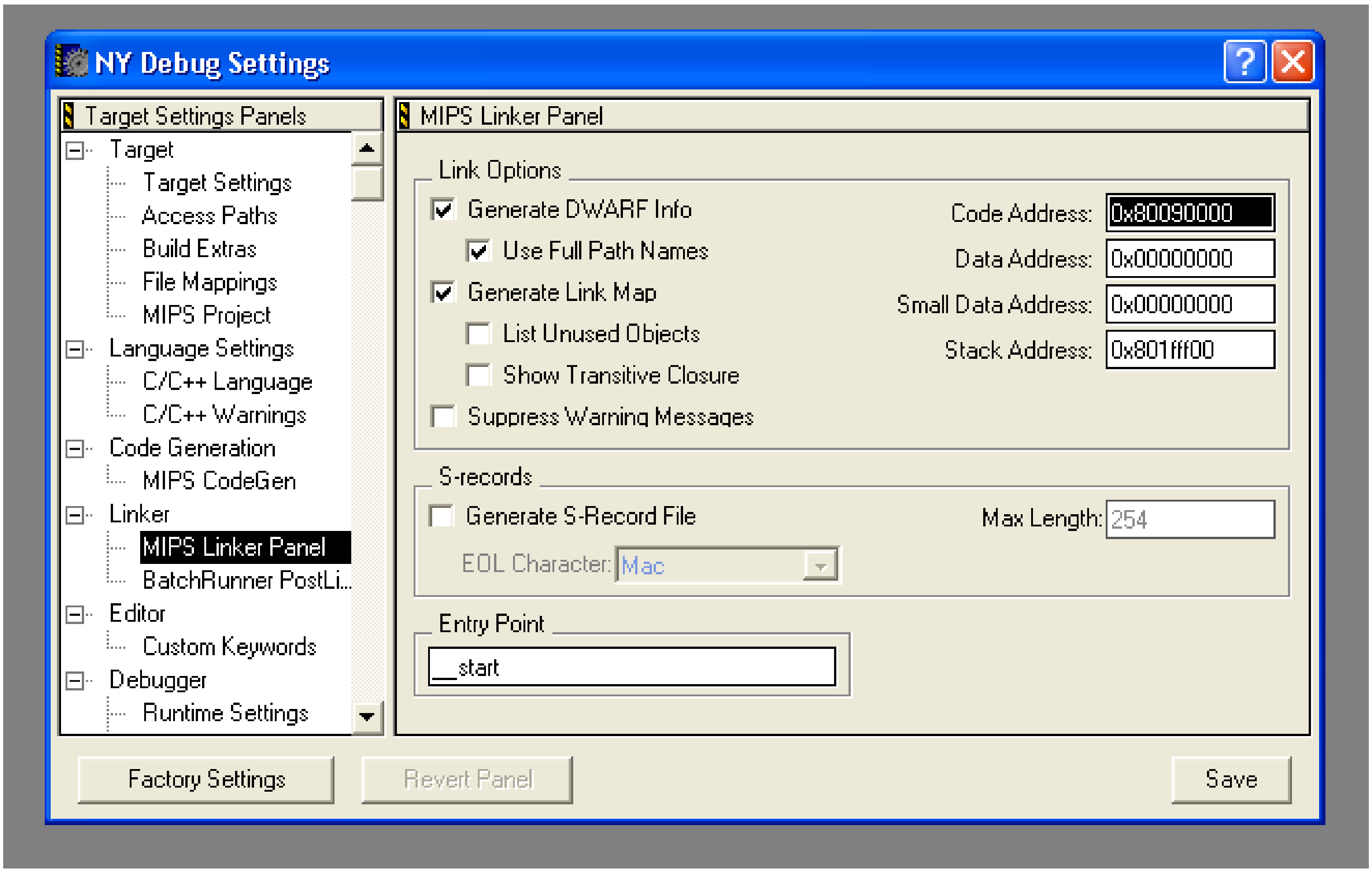
For more information on using CW, read the docs… but a quick run down:

1: Target settings – this is where you configure the compiler (C/C++) and linker options (Address screen below)

Change the “Code Address” field with the link address from the makefile, needs to be done for all the targets.

If you want to debug the target, ie optimized, you must tick Generate DWARF info.
It wont let you launch the debugger without it.

If you use software floats, you must click, Project → add file →(Library files)
Go to folder: C:\Program Files\Metrowerks\CodeWarrior\Net Yaroze Support\Lib
Select file: Math-ISA1-noFPU-LE.lib
Select all targets.


NY Debug Settings

Target Settings Panels
- Target
  - Target Settings
  - Access Paths
  - Build Extras
  - File Mappings
  - MIPS Project
- Language Settings
  - C/C++ Language
  - C/C++ Warnings
- Code Generation
  - MIPS CodeGen
- Linker
  - MIPS Linker Panel
  - BatchRunner PostLi...
- Editor
  - Custom Keywords
- Debugger
  - Runtime Settings

MIPS Linker Panel

Link Options
- [✓] Generate DWARF Info
  - [✓] Use Full Path Names
- [✓] Generate Link Map
  - [ ] List Unused Objects
  - [ ] Show Transitive Closure
- [ ] Suppress Warning Messages

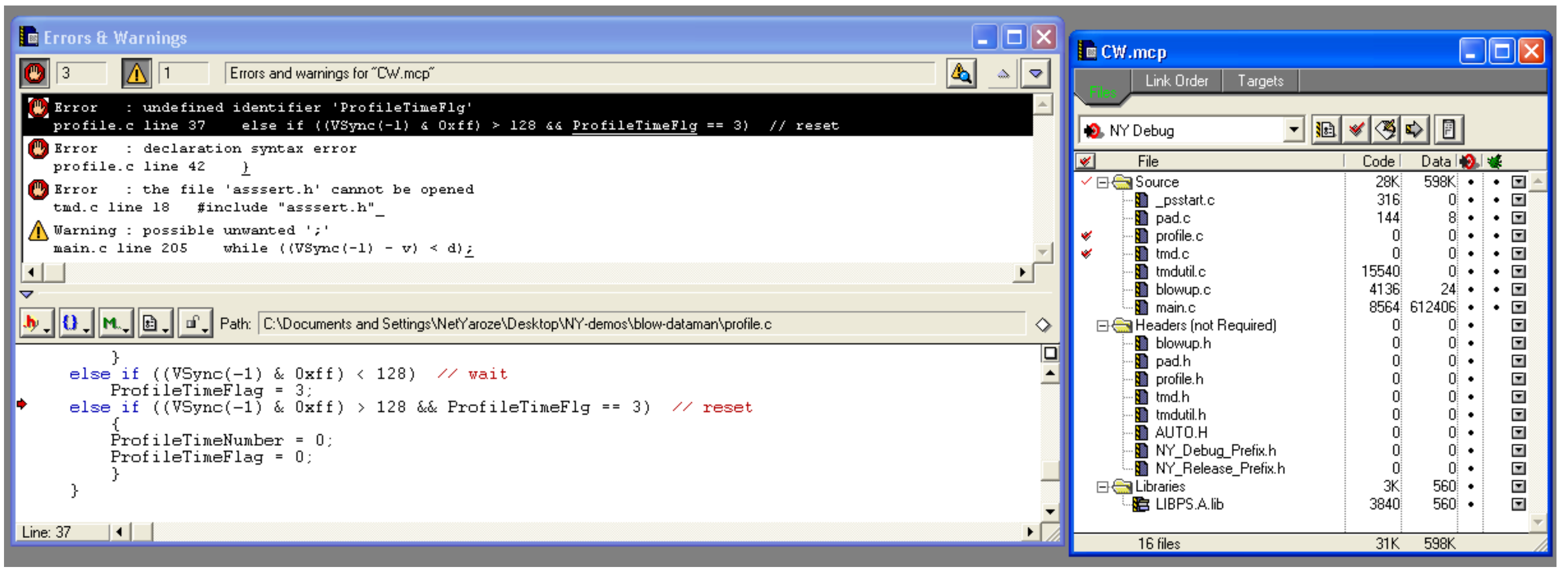
Code Address:	0x80090000
Data Address:	0x00000000
Small Data Address:	0x00000000
Stack Address:	0x801fff00

S-records
- [ ] Generate S-Record File
- Max Length: 254
- EOL Character: Mac

Entry Point: __start

2: Synchronize modification dates
Marks (with red tick) source files on the tree which have changed, ie via an external editor.
But clicking make automatically builds changed files and their associated files.

3: Make – Builds current target
(Left) CTRL – (minus key) to make clean.


Errors & Warnings

3 | 1 | Errors and warnings for "CW.mcp"

```
Error   : undefined identifier 'ProfileTimeFlg'
profile.c line 37    else if ((VSync(-1) & 0xff) > 128 && ProfileTimeFlg == 3)  // reset
Error   : declaration syntax error
profile.c line 42    }
Error   : the file 'asssert.h' cannot be opened
tmd.c line 18   #include "asssert.h"_
Warning : possible unwanted ';'
main.c line 205    while ((VSync(-1) - v) < d);
```

Path: C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman\profile.c

```
        }
    else if ((VSync(-1) & 0xff) < 128)  // wait
        ProfileTimeFlag = 3;
    else if ((VSync(-1) & 0xff) > 128 && ProfileTimeFlg == 3)  // reset
        {
        ProfileTimeNumber = 0;
        ProfileTimeFlag = 0;
        }
    }
```

Line: 37

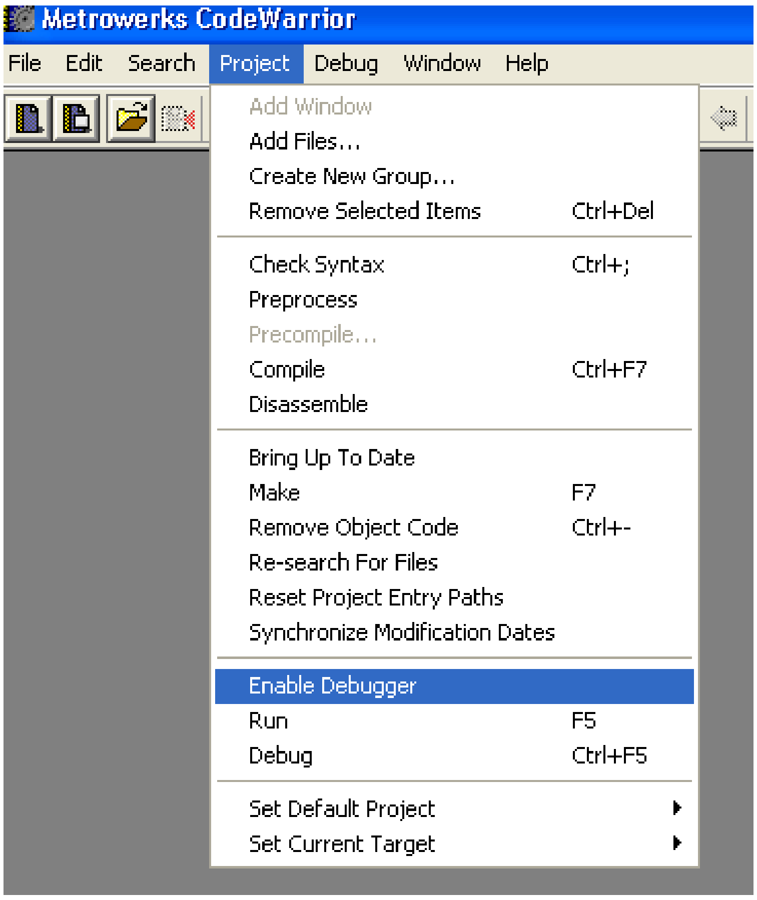
CW.mcp

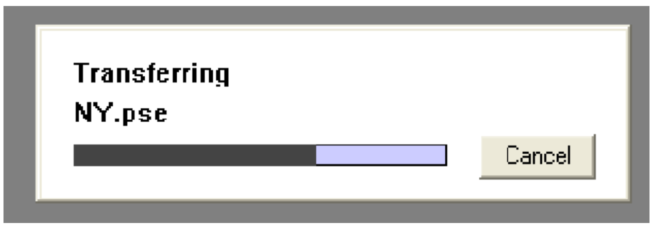
Files | Link Order | Targets

NY Debug

	File	Code	Data
✓	Source	28K	598K
	_psstart.c	316	0
	pad.c	144	8
✓	profile.c	0	0
✓	tmd.c	0	0
	tmdutil.c	15540	0
	blowup.c	4136	24
	main.c	8564	612406
	Headers (not Required)	0	0
	blowup.h	0	0
	pad.h	0	0
	profile.h	0	0
	tmd.h	0	0
	tmdutil.h	0	0
	AUTO.H	0	0
	NY_Debug_Prefix.h	0	0
	NY_Release_Prefix.h	0	0
	Libraries	3K	560
	LIBPS.A.lib	3840	560
	16 files	31K	598K

4: Run/Debug
This will send the debug executable to the playstation via it's internal transfer.
But before you can debug anything, you have to enable it first!





Loading assets before debugging.

Create an executable via yarexe (see below) that exits to siocons before initialising any data, ie before loading TIM's into VRAM etc, the executable will automatically have your data in the correct places in RAM. This is faster and better then sending individual items ie: via CodeWarrior, siocons or unirom/nops.bat etc, then after your program has started siocons, start CodeWarrior’s debugger as normal for debugging.
IE:

```
#ifdef CODEWORRIOR_DEBUGGING
// load exe via unirom/nops and exit to siocons, leaving assets untouched
// then launch in CW debugger

#ifndef __MWERKS__  // if built gcc, load and exit into siocons
        exit_to_siocons();
#endif // CodeWarrior for NY will continue as normal

#endif
```

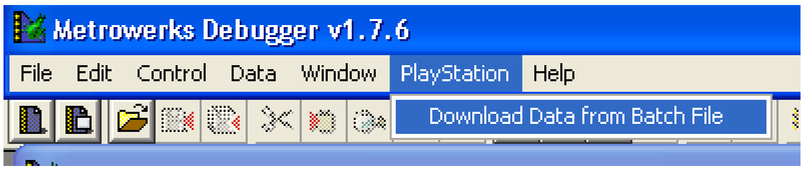
This executable can be burnt to a CDR or sent via unirom and nops.bat, the point is, use the executable to load your assets then exit into siocons, exiting and load siocons will not (generally) effect the existing RAM, then start the CodeWorrior debugger, this is a lot quicker and easier!!!!

See exit_to_siocons below, for an example on out to exit to siocons.

CodeWarrior debugger

It’s that old late 90’s window style ‘free-form’ or ‘multi-window’ mode, and it can not be changed to a a normal Single-Window Mode, and it doesn’t remember your layout!
I recommend returning to the desktop (minimise all apps) and clicking back to the debugger, it’s less confusing.

CW debugger works exclusively in Remote Debug (rdb) mode, so it requires the CDROM to be running in the background (at least at the start).
If you haven’t already loaded your assets into RAM, run menu:
Playstation → Download Data from Batch File:



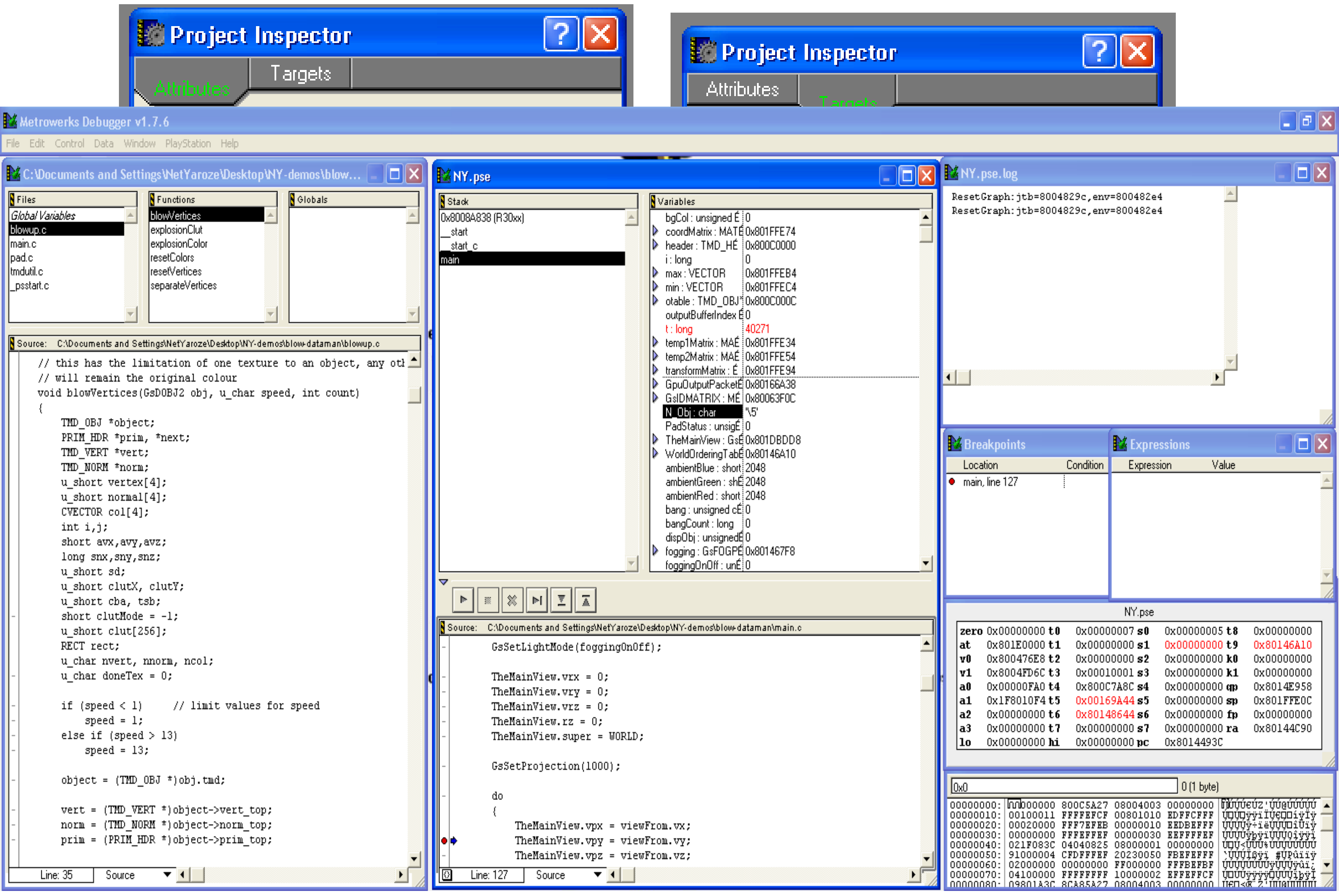
This loader wants a *.sio file where as the CW PSComUtil defaults to a *.dat, but it’s the same thing, an AUTO batch file that loads the assets, the load executable and go statements are ignored.

There is no scroll wheel mouse support on debug windows!
To circumvent this from C:\misc you can install the freeware Fly Wheel app, I recommend removing it from the start → all programs → startup and manually starting it when debugging, it’s buggy but works, but your mileage may vary.

Step over:	S + CTRL
Step into:	T + CTRL
Step out:	U+ CTRL
Run:	R+ CTRL

5: Project inspector

Here pick some options assisted to each file and pick which target each file is used in.



Files: Global Variables, blowup.c, main.c, pad.c, tmdutil.c, _psstart.c

Functions: blowVertices, explosionClut, explosionColor, resetColors, resetVertices, separateVertices

Source: C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman\blowup.c

```
// this has the limitation of one texture to an object, any ot
// will remain the original colour
void blowVertices(GsDOBJ2 obj, u_char speed, int count)
{
    TMD_OBJ *object;
    PRIM_HDR *prim, *next;
    TMD_VERT *vert;
    TMD_NORM *norm;
    u_short vertex[4];
    u_short normal[4];
    CVECTOR col[4];
    int i,j;
    short avx,avy,avz;
    long snx,sny,snz;
    u_short sd;
    u_short clutX, clutY;
    u_short cba, tsb;
    short clutMode = -1;
    u_short clut[256];
    RECT rect;
    u_char nvert, nnorm, ncol;
    u_char doneTex = 0;

    if (speed < 1)      // limit values for speed
        speed = 1;
    else if (speed > 13)
        speed = 13;

    object = (TMD_OBJ *)obj.tmd;

    vert = (TMD_VERT *)object->vert_top;
    norm = (TMD_NORM *)object->norm_top;
    prim = (PRIM_HDR *)object->prim_top;
```

Line: 35 Source

Stack:

```
0x8008A838 (R30xx)
__start
__start_c
main
```

Variables	
bgCol : unsigned É	0
coordMatrix : MATÉ	0x801FFE74
header : TMD_HÉ	0x800C0000
i : long	0
max : VECTOR	0x801FFEB4
min : VECTOR	0x801FFEC4
otable : TMD_OBJ*	0x800C000C
outputBufferIndex É	0
t : long	40271
temp1Matrix : MAÉ	0x801FFE34
temp2Matrix : MAÉ	0x801FFE54
transformMatrix : É	0x801FFE94
GpuOutputPacketÉ	0x80166A38
GsIDMATRIX : MÉ	0x80063F0C
N_Obj : char	'\5'
PadStatus : unsigÉ	0
TheMainView : GsÉ	0x801DBDD8
WorldOrderingTabÉ	0x80146A10
ambientBlue : short	2048
ambientGreen : shÉ	2048
ambientRed : short	2048
bang : unsigned cÉ	0
bangCount : long	0
dispObj : unsignedÉ	0
fogging : GsFOGPÉ	0x801467F8
foggingOnOff : unÉ	0

Source: C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman\main.c

```
    GsSetLightMode(foggingOnOff);

    TheMainView.vrx = 0;
    TheMainView.vry = 0;
    TheMainView.vrz = 0;
    TheMainView.rz = 0;
    TheMainView.super = WORLD;

    GsSetProjection(1000);

    do
    {
        TheMainView.vpx = viewFrom.vx;
        TheMainView.vpy = viewFrom.vy;
        TheMainView.vpz = viewFrom.vz;
```

Line: 127 Source

```
ResetGraph:jtb=8004829c,env=800482e4
ResetGraph:jtb=8004829c,env=800482e4
```

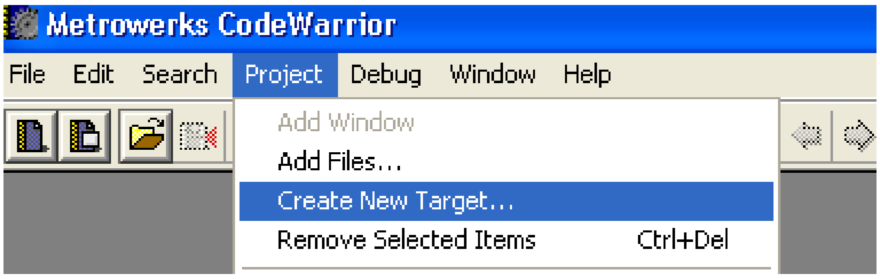
Location	Condition
main, line 127	

```
zero 0x00000000 t0 0x00000007 s0 0x00000005 t8 0x00000000
at   0x801E0000 t1 0x00000000 s1 0x00000000 t9 0x80146A10
v0   0x800476E8 t2 0x00000000 s2 0x00000000 k0 0x00000000
v1   0x8004FD6C t3 0x00010001 s3 0x00000000 k1 0x00000000
a0   0x00000FA0 t4 0x800C7A8C s4 0x00000000 gp 0x8014E958
a1   0x1F8010F4 t5 0x00169A44 s5 0x00000000 sp 0x801FFE0C
a2   0x00000000 t6 0x80148644 s6 0x00000000 fp 0x00000000
a3   0x00000000 t7 0x00000000 s7 0x00000000 ra 0x80144C90
lo   0x00000000 hi 0x00000000 pc 0x8014493C
```

0x0 0 (1 byte)

```
00000000: 00000000 800C5A27 08004003 00000000
00000010: 00100011 FFFFEFCF 00801010 EDFFCFFF
00000020: 00020000 FFF7EFEB 00000010 EEDBEFFF
00000030: 00000000 FFFEFFEF 00000030 EEFFFFEF
00000040: 021F083C 04040825 08000001 00000000
00000050: 91000004 CFDFFFEF 20230050 FBEFEFFF
00000060: 02000000 00000000 FF000000 FFFBEFBF
00000070: 04100000 FFFFFFFF 10000002 EFFEFFCF
00000080: 09801A3C 8CA85A27 08004003 00000000
```

Targets can be used to work on new/old code base for example, you can create new targets via: Project → Create New Target...

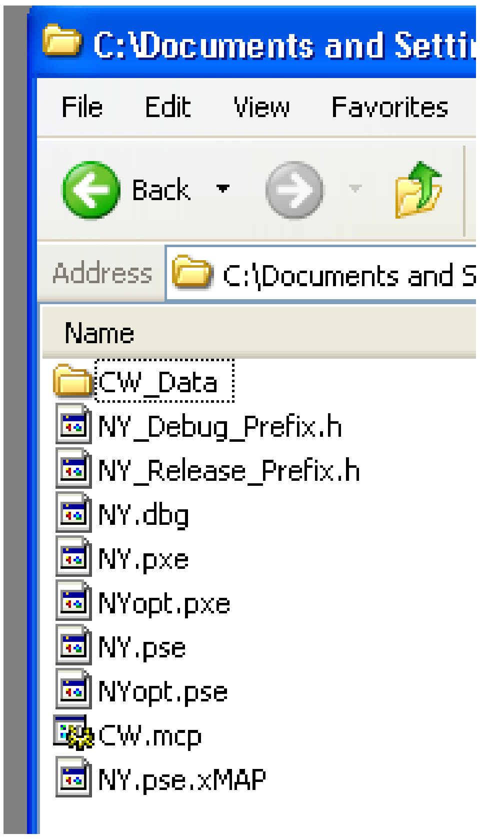


CodeWorrior executable files

Each target produces the following files:

.pxe – CW executable format.

.pse – CW debugging format.



CW_Data
NY_Debug_Prefix.h
NY_Release_Prefix.h
NY.dbg
NY.pxe
NYopt.pxe
NY.pse
NYopt.pse
CW.mcp
NY.pse.xMAP

Using CW Executables in emulators

CodeWarrior uses the same siocons auto batch loading text file.

Where the common AUTO file loads the GNU GCC executable like:
local load main

This line is replaced with your CW PXE file (not the PSE debugging format):
local load CW\NYopt.pxe

Where CW is the project name, with no spaces etc.
The CW auto file ie: cw.auto can be used with PCSXR and emulators that can load PS-X Executables.

PCSXR

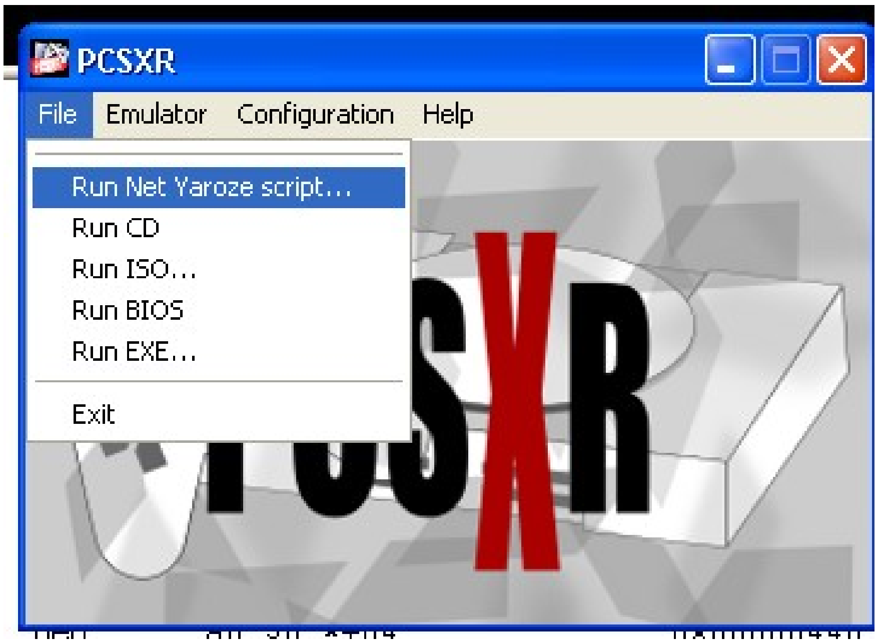
This PCSXR build has been configured to load Net Yaroze scripts, including CodeWorrior executables.

This version of PCSXR is a lot older and tends to be slow and buggy, it has no TTY output but executable has access to the CDROM, Command line:

pcsxr.exe -yaroze "FULLPATH\AUTO" (must have quotes around full path and auto file)

Example:
pcsxr. -yaroze "C:\Documents and Settings\NetYaroze\Desktop\NY-demos\blow-dataman\cw.auto"

Or via GUI via: File → Run →Net Yaroze script...



Yarexe

Combines, libps.exe with assets and CW executable into a single PS-X executable.

This can then be used with any emulator (or sent to hardware or burnt to a CDROM).
Run:
yarexe CW.AUTO

This will output: psx.exe

It can then be loaded in no$psx, however, there's no access to the NY boot CDROM.

If you are sending this file to a playstation, (ie via nops) it can be compressed (for a lot faster transfer speeds!) by:
upx –best psx.exe

Both apps create PS-X executables from the CW pxe file, PCSXR automatically and yarexe will leave it with the -v option.

CodeWorrior Code Tips

exit_to_siocons

```
void exit_to_siocons(void)
{
   /* Matt Verran's code from
      Subject: loading an exe and using Exec()
      Date: Sun, 15 Apr 2001 10:23:17 +0100
      From: Matt Verran
      Newsgroups: scee.yaroze.freetalk.english
    */

   if( CdSearchFile(0, "\\DTL_S30.35;1") == 0 )
   {
      printf("\n DTL_S30.35;1 not found \n");
   }
   else
   {
      struct EXEC  *exec_struct;
      s32 result;

      printf("\n DTL_S30.35;1  found, de init \n");

      ResetGraph(3);


      printf("\n Loading DTL_S30.35;1\n");

      exec_struct = CdReadExec( "\\DTL_S30.35;1" );
      result = CdReadSync(0, 0);

      printf("\n found loaded, result:  %d    \n\n", result);

      if (result==0)
      {


         printf("\n executing \n");
         EnterCriticalSection();
         Exec(exec_struct, 0, 0);
         ExitCriticalSection();
      }
   }

}
```

CodeWorrior Defines

```
#ifdef __MWERKS__ // CodeWarrior for NY precompiler define
#define __FUNCTION__ "CodeWorrior does not support the __FUNCTION__ macro!"
#endif // CodeWarrior for NY
```